SLIDE 2

Hi. I'm Tracy Abrahms. I'm a Junior Javascript Engineer on the Web team at Urban Airship. You can find me in most places online by my handle hackygolucky.

SLIDE 3

A funny thing happened when I went rock climbing...every other person I'd jump on a rock next to wound up being a programmer.

Go top-roping at Portland Rock Gym, run into a Javascript Admirers meetup friend. Head over to The Circuit, and there are two fellow PyLadies bouldering away.

A climbing buddy of mine said it best, "the reason we programmers enjoy climbing is that we love problem-solving and we can't turn it off. So, we put ourselves into climbing and 30 feet up, and your mind and body have no choice but to focus on a much different type of problem."

SLIDE 4

Eleven months ago, I decided to quit my job, change career paths, and become a professional problem-solver. Running into these like-minded individuals was confirmation of a path I was stoked to be on. Maybe confirmation of a little neuroses as well.

SLIDE 5

I dove in head first. What could I do to learn programming? Going back to school for a degree was my first thought. A computer science degree is a very thorough but slow way to achieve this.

But, I mean, "punching rocks". I don't steer away from pain. Why remove the tortured struggle of having to teach yourself? Taking on another student loan, well, that fiscal math was hard to swallow.

Another option I found was the code bootcamps popping up. After 12 weeks, you too can become a junior developer! Odd thing was, of the eight schools I could find, only one taught Javascript as a primary focus. All others used Ruby. Nothing against Ruby.

SLIDE 6

Some really great advice I got when I was weighing in on how to get started, was to find a community. Huh? I wasn't sold on this. I'm independent. I don't need a group of people teaching me how to code. Don't all coders just sit in a poorly-lit room late into the night cranking away projects alone? Alone? Not even close. I was told to look for the best community you had access to, and learn the language. I was surrounded by friends in Python, so off I dove into a coding circle, meetups, and workshops.

And got spoiled. I don't know about many communities. I haven't been in them. I have been in Python. They have exhaustively worked out really great solutions for welcoming new programmers with open arms. Or it might just be programmers here in Portland. Must be something in the water, other than fluoride of course.

Now a community is a huge help in rock climbing as well. You can't go climbing outside alone. It isn't safe. You need a spotter or a belay partner if you're rope-climbing.

Even moreso, you need the brains of other climbers. All the knowledge in their noggins. That's valuable. That'll keep you safe and having fun.

When I look around everyday at work and think of the collective knowledge and experience that's in the brains of my colleagues, I smile and take a deep breath. I have a very long journey ahead of me.

SLIDE 7
So what did I start cramming my noggin with? In climbing, there's little you can do without an intro class. You don't have to be fit. And as crazy as it may sound, I did try to read books to learn technique.But you need to know the basics of staying on a wall. It would be like trying to program without knowing how to use a keyboard. You can fumble your way through it, but you're gonna have a bad time.

So I turned to my friend, the internet. And apparently, many others have too. MOOCs(massive online open courses) like Coursera began their courseload with classes like CS101, and Kahn Academy as well as many others are in the game. CodeAcademy, a much more focused online coding tutorial tool, allowed me to learn how to program in-browser and avoid that whole pesky 'also have to learn a dev environment' thing. You know, distractions.

While it gave me decent practice with vanilla javascript, html, and css--I in no way learned how to manage the pieces of a stack. To be fair, I didn't know what that phrase meant. Well that, my dears, is what books are for! The public library and all of the programmer books you can fathom, are great references. They aren't, however, up to date. So when I was introduced to the idea of 'strict mode' far later than I should have been(agree with it, OR not), I got a nice wake up call that programming books can age-out.

Sort of the opposite of that 3-year old book on javascript is technical blog posts. Tutorials and documentation.

Node is pretty new. It's hard to find a crusty source of info on it. It's more likely to become obsolete because of how swiftly the Node world moves. I found in this, that there are so many really great resources. HowtoNode by Tim Caswell, Javascriptissexy by Richard Bovell, Substack's Streams handbook, the Art of Node by Max Ogden is fantastic...I'm track weekly to see if there are updates on it. I want other people to learn Node too!

Nodejs.org API docs are beautiful. To be honest. Clean and concise. I'm still trying to catch up with all the past NodeUp episodes on my walks to work. What I have struggled with as a fairly new programmer, however, is that I need more how to's in Nodejs. Streams are awesome. They're also heady(for me). Event-based programming is something that some people in higher level languages shudder at the thought of. There are so many sources of truth to go to as a new programmer in Nodejs, it can be overwhelming. I'm not saying Nodejs.org needs to be MDN. I can say at times I wished it was, however.

And like a safety check before I start climbing, I'll review--my resources all supplemented each other: free online courses, in person workshops, programming books, and online tutorials focused on my set short and long-term goals.

**I hesitate to recommend any one book. Some work for a certain learning type. Be flexible and try out another book if you're having trouble. Mutltiple perspectives can be a little confusing, but can help hone questions you're having.

SLIDE 8
So standing armed with my spirit guides and my hazardous amount of knowledge, a serendipitous moment occurred where a really great employer was hiring and just happened to want someone who was excited to break lots of things.

No, actually, my company was willing to offer me an opportunity to work as a junior javascript developer and learn from a rock-awesome mentor while gaining experience in a full dev environment. No, I did not commit code to the production environment in the first work week. I need no bragging rights in that. I'm still allowed to work there. That's thanks enough.

But this did and does give me the most able way possible to...

SLIDE 9
BREAK EVERYTHING.

Well, not quite

But you should try? Practice is key! As a kid, I always dove in. I loved to climb trees and jumped into calculus. Learning through experience and my subsequent mistakes made my lessons visceral and immediate. As I got older, I became a little more willing to allow others to make these moves and watch them falter or flail.

This is why I love rock climbing. I'm terrified of heights. I'm not kidding. There are days when I'm inside practicing and I'm physically shaking on the wall. I'm totally psyched out. That's okay. Challenging myself is good. And my shakey days are fewer in-between.

This is why I love coding. I'm still struggling with this as a programmer. Should I open source my project? Where do I even start with this feature? Just code, darn it. Then fix it. Then break it again. Get something on the page. It's not doing anything when it's in your head!

So project based learning is where it's at. In climbing, a project is something you have to keep coming back and working on. It's a climb you're really going to struggle with. And you'll eventually get it.

This. THIS is why I have very little skin left on my knuckles. When I'm fully-tuned into my climb, and my reflexes are allowing me to think of only what is next in my hold sequence, I'll go for a hold. I'm short. I have to get a bit reachy. When I do this, it's full force. Unfortunately, my aim might be off. Instead of grabbing the hold I intended, I'm punched a rock that was much closer and inconveniently in the way. It hurts a bit. It's only a little skin.

SLIDE 10
Which leads me to: Write it all down. You're typing all day. It's okay to write for a little bit. Write down everything you aren't understanding. Write down your big and small victories. Write down all the wild ideas for projects you can't wait to start. This will be a great gauge to see how you're

doing, remember questions to ask people you can't address right this second, and allow you to look back later on and laugh that you had no idea what a stream was. Likewise, in climbing, logging your gear purchase dates and the big falls they have taken is essential. That'll help you decide when to retire your worn out rope.

I have written down so many questions since I've started. So many words I didn't have time to look up.

SLIDE 11
The first instruction you'll receive when you are learning to climb is your safety check. You, the climber, and your partner, the belayer, must check each other's equipment and redundancies to ensure you are both safe.

Then you signal. "On belay?" If you're partner is ready, "Belay on". "Climbing"(step up to your problem). Everyone is set. When fully prepared, 'climb on!'

Language is innately purposeful in programming. This can be hugely problematic for someone career transitioning.
library. Okay. Object. Well yes, thank you, that's super vague. API. What kind? parse. lexing. side effects. listener. These all have very specific and LOADED meanings. There are chapters of books, heck whole books based on these words. No one warned me of this. Really hard days are when you don't have the words to articulate your problem. That's a fun epiphany the day you realize that variables are actually supposed to communicate a purpose and not just be cute. Programmers are clever. I, apparently, can be dense.

I have an ever-growing list of terms used, abused, and mostly mangled by me that I have acquired through my short stint thus far.

Mis-using terminology is only the start of my transgressions...

SLIDE 12
I'm such a hazard. There are many days where I will ask a question and get a very odd response. Why is it odd? Because I knew so little about the problem, that I didn't even know what I didn't know. And fortunately for me, the person responding understood this. Yikes. I owe my teammates and significant other drinks for life.

I need to work on focus at every level. From my email distractions to not choosing a side project to stick to, I need to give myself a better fighting chance of growing some really great programmer habits.

I'm still being very impatient. I want to learn it all. All of it. Now.

I still hesitate to ask questions. My brain is still trying to process what's been said and somehow my head involuntarily nods 'yes' when asked if I understood. 5 minutes later, 'what in the world was he talking about? Crap.'

I could also probably try getting a little more sleep.

SLIDE 13
I'm here. That's not me kissing butt. I'm not a fan of public speaking. But I've worked hard for the changes I've made in my life over the past few years, and if I were to stop challenging myself now, it'd all be for naught. Even moreso, I feel a responsibility to share my road thus far in the hopes that I can make it an awesome path for others to follow. A few people really went above and beyond to help me into the programming community, and it is a blessing I am happy pass on.

I'm meeting others who have the tenacity and curiosity to be a programmer. And they were just never introduced to programming. This is an obsession I'm happy to share. So I am.

SLIDE 14
A fellow rock-addicted programmer said,
" Sometimes to get to the next level in climbing, you need to go outside of it. You may need to take up yoga for flexibility or weightlifting for strength. Likewise, in programming, sometimes you need to go sideways to go up. If you want to master Node.js, you'll probably want to spend time outside of it and work on your command line and version control skills. Programming has a huge amount of silos. To go very deep in any one you have to gain competence in many."

Node.js is a special beast. This wonderful back-end world that makes Javascript really FUN. In order to understand it better, I have to continue working on more expanding my skillset. One of my favorite projects coming up...

SLIDE 15
Remember when I mentioned the Python community earlier? I'm still a part of it. I attended PyCon this year. Jessica McKellar of OpenHatch.org talked about growing the Boston Python user group to 1700 members through workshops....and I trolled the heck out of them. "But how do you think I could make a workshop like this for Nooooodejs? How could I get people excited like this for Node?" Yeah, I got a few eye rolls. A speaker on the panel was a Ruby organizer who had gotten assistance from the Python community. When I asked for his advice, he looked like I was going to give him Node cooties.

I help Ben and Merlyn organize the local Node.js user group, and we could use more Node.js friends! I'm really excited about helping to organize our local Nodejs user group here in Portland. I'm already brainstorming with others as to how we can build a 'Welcome to Nodejs' Nodecopters workshop to get more people excited about Nodejs, AND make it accessible to those who are just getting started. One of the earlier projects I created in Node was an irc bot, and I hope to polish that up and get it out as a tutorial as well. Ultimately, I'd love to document these workshops so that other user groups can get people learning Nodejs and hook them into the community.

Working on challenges, in both rock climbing and coding, is a life I'm happy to share. If you're interested, you should join us at a monthly PDXNode meetup. If you're interested in working out a little of your keyboard aggression on some rocks, I know a few places. Either way, roughing up your hands as part of the learning process.

I had considered rapelling down from the ceiling, but the ascension post-talk would have been REALLY awkward while the next speaker was on. Would have taken like 15 minutes. Thanks everyone. You are kind.